PC503 Programming Lab
Insem II Exam

Q1. Write a function analyze_array that takes the size and elements of an array. Create a Numpy array and print a dictionary containing the following statistics: Mean Median Variance Standard Deviation 25th Percentile 75th Percentile

Input Format

1. The first line contains an integer n (1 ≤n ≤ 100), the number of students.
2. A second line containing space-separated floating-point numbers representing the elements of the array.

Constraints

The input list will contain only floating-point numbers. The length of the array will not exceed 1000 elements.

Output Format

Return a dictionary with the computed statistics rounded to two decimal places. The keys of the dictionary should be: "mean" "median" "variance" "standard_deviation" "25th_percentile" "75th_percentile"

Sample Input 0

```
1.5 2.5 3.5 4.5 5.5
```

Sample Output 0

```
{
  "mean": 3.5,
  "median": 3.5,
  "variance": 2.5,
  "standard_deviation": 1.58,
  "25th_percentile": 2.0,
  "75th_percentile": 5.0
}
```

Q2. Write a function count_unique_substrings(s)that takes a string s as input and prints the number of unique substrings in s.

Input Format

A single line containing the string s (1 ≤ |s| ≤ 1000), where |s| is the length of the string. The string consists of only lowercase English letters.

Constraints

The input string will contain only lowercase English letters. The length of the string will not exceed 1000 characters.

Output Format

Print an integer representing the number of unique substrings of the string s.

Sample Input 0

```
abc
```

Sample Output 0

```
6
```

Explanation 0

The unique substrings are: "a", "b", "c", "ab", "bc", "abc".

Q3. Implement a student grade management system using a class and objects. The system should be able to record student names and their scores in multiple subjects, then calculate and print the average grade for each student. If a student’s average grade is below a certain threshold, they are considered to have failed.

Input Format

3. The first line contains an integer n (1 ≤n ≤ 100), the number of students.
4. The next n lines each contain:
   - A string name (1 ≤ |name| ≤ 100), the name of the student.
   - A space-separated list of scores s1, s2,...,sk (0 ≤ k ≤ 10), where each score si (0 ≤ si ≤ 100) is an integer representing the score the student received.

Constraints

- Names will contain only alphabetical characters.
- There can be up to 10 scores per student.
- The scores will be integers between 0 and 100 inclusive.

Output Format

For each student, print the name followed by their average score. If the average is below 60, append "(Fail)" to their name. If they pass, just print their name and average grade.

Sample Input

```
3
Alice 80 90 85
Bob 50 60 55
Charlie 70 80
```

You may assume that each student has at least one score

Sample Output

```
Alice 85.00
Bob 55.00 (Fail)
Charlie 75.00
```

Ensure to format the average grade to two decimal places.

Explanation

- Alice's average is (80+90+85)/3=85.00(80 + 90 + 85) / 3 = 85.00(80+90+85)/3=85.00
- Bob's average is (50+60+55)/3=55.00(50 + 60 + 55) / 3 = 55.00(50+60+55)/3=55.00 which is below the threshold, so he fails.

- Charlie's average is (70+80)/2=75.00(70 + 80) / 2 = 75.00(70+80)/2=75.00

Q4 You are given a dataset representing sales of various products across different regions. Your task is to write a function that analyzes the sales data and prints the average price of the top-selling products in each region. The input will be provided as a list of dictionaries, each representing a sale. Each sale contains the following information:

- **Product Name** (string)
- **Price** (float)
- **Region** (string)
- **Quantity Sold** (int)

Your function should calculate the total sales for each product in each region and then determine the average price of the products that sold the most units in each region.

Input Format

- A list of dictionaries where each dictionary represents a sale.
- Each dictionary contains:
  - "Product Name": (string) Name of the product.
  - "Price": (float) Price of the product.
  - "Region": (string) The region where the product was sold.
  - "Quantity Sold": (int) The quantity sold of the product.

Output Format

- A dictionary where each key is a region and the value is the average price of the top-selling product in that region.

Constraints

- The number of sales entries will not exceed 10510^5105.
- The price will be a positive float with at most 2 decimal places.
- The quantity sold will be a positive integer.
- The product names and regions will be non-empty strings.

Sample Input:

```
sales_data = [
  {"Product Name": "Product A", "Price": 20.0, "Region": "North", "Quantity Sold": 30},
  {"Product Name": "Product B", "Price": 15.0, "Region": "North", "Quantity Sold": 50},
  {"Product Name": "Product A", "Price": 20.0, "Region": "South", "Quantity Sold": 40},
  {"Product Name": "Product C", "Price": 25.0, "Region": "South", "Quantity Sold": 20},
  {"Product Name": "Product B", "Price": 15.0, "Region": "East", "Quantity Sold": 25},
  {"Product Name": "Product D", "Price": 30.0, "Region": "East", "Quantity Sold": 30}
]
```

Sample Output:

```
{
    "North": 15.0,
    "South": 20.0,
    "East": 30.0
}
```

Q5 You are supposed to analyze stock prices over a given number of days. The user will input the number of days and the stock prices for each day. Your function should create a DataFrame from this input, perform filtering based on specified criteria, and print a summary of the stock prices.

Input Format

1. **First Input:** An integer days representing the number of days (1 ≤ days ≤ 100).
2. **Second Input:** A list of floats prices representing the stock prices for each day.

Output Format

A pandas DataFrame containing only the days where the stock price was greater than the average.

Constraints

- The length of the prices list will be equal to days.
- All stock prices will be positive floats (0 < price ≤ 10,000).

Sample Input

```
days = 5
prices = [100.5, 200.75, 150.0, 300.0, 250.25]
```

Sample Output

```
   Day  Stock Price
0   2      200.75
1   4      300.00
2   5      250.25
```